Geosci. Model Dev. Discuss., 6, C805–C809, 2013
www.geosci-model-dev-discuss.net/6/C805/2013/

Geoscientific Model Development Discussions
Open Access

Interactive comment on "Parallel algorithms for planar and spherical Delaunay construction with an application to centroidal Voronoi tessellations" *by* D. W. Jacobsen et al.

D. W. Jacobsen et al.

jacobsen.douglas@gmail.com

Received and published: 18 June 2013

We would like to thank reviewer #1 for their useful comments. Responses are interleaved with originally reviewer comments.

———— Reviewer comment ——

The paper "Parallel algorithms for planar and spherical Delaunay construction with an application to centroidal Voronoi tessellations" presents a process for constructing 2D Delaunay triangulations in parallel of a set of points (generators) on the sphere. The focus of the method is the construction of a centroidal Voronoi tessellation. The ap- proach is to create overlapping subdomains (speciïňĄcally circles), in which two

regions are identiïňĄed: a) a region whose local Delaunay triangulation is also globally Delaunay, b) a region where the global Delaunay property is not guaranteed, but it is covered by the regions a) of the neighboring subdomains. While the approach is very interesting, the presentation has some major gaps and my recommendation is to be reconsidered after major revisions.

The core of the algorithm lies on decomposing the domain (the sphere surface) into overlapping circles. The generators included in the circles are projected to the plane where Triangle is used for creating the triangulation. The set of triangles T whose circumcircle is inside the subdomain circle is also globally Delaunay, while the rest triangles are discarded from the triangulation. The basic requirement for the correctness of the algorithm is:

1. The union of T gives the global Delaunay triangulation.

Other considerations that should be taken into account (but are not requirements) are:

2. The overlapping areas should be minimum in order to avoid unnecessary cost.

3. The decomposition should be reasonably balanced.

The authors propose the radious of the circles to be the max distance of the center to the neighbor's centers (page 1444, 3-5), but they do not provide any formal proof of the 1. condition. The size of the subdomains, as well as the size of the overlapping areas, depend on the local size of the Delaunay triangle radius. Especially for non-uniform tessellations this maybe challenging. One needs not to cover all cases, reasonable assumptions on the nature of the grid and the density function can be used, such as smoothness. The convergence properties of CVT construction can also be of help. The approach of utilizing an initial Voronoi diagram as basis for constructing the decomposition is a good start. The derived bounds of the sizes may help also for reducing the overlapping areas, which currently triple the triangulation load (if my counting is correct). It can also provide some clues for better load balancing. While degenerate cases can be safely ignored, some discussion on cases where the method may not work should be included.

—————— Author Response —————-

We attempted to come up with a formal proof for the radial distance, but every version we came up with had a case where it would break. Alternatively, we came up with some sort methods that should guarantee global Delaunay, but they are very inefficient when coded properly. When tested with the grids presented in the paper they produce identical results. We have plans to try and release a version of our sort algorithm that guarantees global Delaunay properties, but so far it has yet to be implemented in a satisfactory way.

We did not like the idea of adding a proof which was easily broken, so instead we decided to focus the paper on the alternative sort method (described Voronoi sort), rather than the distance based sort. Along with this, some discussion about when a decomposition wouldn't work is given as well, to try and aid cases where the sort/decomposition method would break down. These two coupled together should be sufficient for guaranteeing global Delaunay in most cases.

We also listed the three criteria a sort method needs to satisfy to be valid globally, both for global Delaunay triangulations, and for SCVT generation. And as soon as the guaranteed sort method is sufficiently coded, we will release that for use in both cases as well.

—————— Reviewer Comment ——————

The experimental results also need some clariïňĄcation. In Table 1 performance results in comparison with STRIPACK for constructing a single Delaunay triangulation are given. It appears that the proposed method is signiïňĄcantly slower, while Triangle (which is used as far as I understand by MPI-SCVT) is probably the fastest 2D Delaunay triangulator.

—————- Author Response ———————-

Some recent improvements to how MPI-SCVT handles the merge step have been included in the results tables.

Previous versions made use of the TR1 C++0x unordered_set for determining uniqueness of triangles, but this has been removed in the more recent versions.

Although the proposed method is still slower for 1 and 2 processors, it is now faster when using 96 processors. The speedups gained also increase as a function of generator count. These can be seen in the updated results figures.

—————- Reviewer Comment ———————

In page 1448, lines 1-2, the authors indicate that the merge step is the one that is costly, but it is not clear to me where this cost comes from (especially for an algo- rithm designed to minimize the merge cost). It may also be a matter of implementation.

—————- Author Response ———————-

This has been removed due to the changes included from the previous response.

—————- Reviewer Comment ———————

Again, in Table 4, STRIPACK triangulation outperforms MPI-SCVT (on one processor) by a factor of 10, but the total iteration is almost 10 times slower. Is there a reason for the Loyd iteration to be so much slower when using STRIPACK? Or is there another part that slows it?

—————- Author Response ———————-

This was an issue with the default values of the STRIPACK generator. As a result, the generation tests have been re-run, and the results have been regenerated.

The new results also include improvements to MPI-SCVT, which allow the full triangulation routine to be more competitive versus STRIPACK.

—————- Reviewer Comment ——————

In page 1443, line 21, a cluster with "6176 cores per node" is cited; is this correct?

—————- Author Response ———————-

No, there are actually 24 cores per node, where the node utilizes AMD Opteron 6176 chips. The sentence has been updated to make it more clear.

—————- Reviewer Comment ——————

Finally, if the software is publicly available, it would be useful to cite it in the paper.

————— Author response ————————

Thank you for the suggestion. A citation has been added.

Interactive comment on Geosci. Model Dev. Discuss., 6, 1427, 2013.